



⑬ BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENTAMT

⑫ **Offenlegungsschrift**
⑩ **DE 196 25 841 A 1**

⑤① Int. Cl.⁸:
G 06 F 19/00
A 61 B 5/055
A 61 B 6/03
A 61 B 8/12
// G06F 159:00

DE 196 25 841 A 1

②① Aktenzeichen: 196 25 841.3
②② Anmeldetag: 27. 6. 96
④③ Offenlegungstag: 2. 1. 98

⑦① Anmelder:
Siemens AG, 80333 München, DE

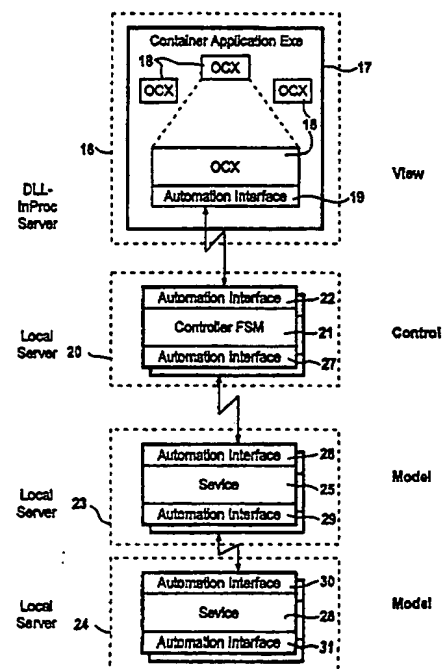
⑦② Erfinder:
Dorn, Karlheinz, Dipl.-Inform., 90562 Kalchreuth, DE;
Becker, Detlef, Dipl.-Ing., 91096 Möhrendorf, DE;
Quehl, Dietrich, Dipl.-Ing., 91052 Erlangen, DE

⑤⑥ Entgegenhaltungen:
MORNEBURG, Heinz (Hrsg.): Bildgebende Systeme für die medizinische Diagnostik, Erlangen, Publics MCD Verlag, 3. Aufl., 1995, S. 680-697;
LANG Eric: Software aus OCX-Komponenten, Microsoft Systems Journal 1/95, 1995, S. 25-38;
ORFALI, Bob, HARKEY, Dan: Client/Server with Distributed Objects, Byte, April 1995, S. 151-162;

Prüfungsantrag gem. § 44 PatG ist gestellt

⑤④ Medizinische Systemarchitektur, basierend auf Microsoft OLE/OCX und Automation, bzw. Atomic

⑤⑦ Die Erfindung betrifft eine medizinische Systemarchitektur mit einer Modalität (1 bis 4) zur Erfassung von Bildern, einer Vorrichtung (5 bis 8, 11, 12) zur Verarbeitung der Bilder und einer Vorrichtung (9) zur Übertragung der Bilder, bei dem die Vorrichtung (5 bis 8, 11, 12) zur Verarbeitung ein digitales Bildsystem mit einem Rechner aufweist, der nach einem Verfahren zum Datenaustausch zwischen verschiedenen Anwendungsprogrammen (OLE) mit grafischen Steuerelementen und einem Standard für OLE Custom Controls (OCX) arbeitet, wobei jedem einzelnen durch Grenzen limitierten Prozeß ein OCX-Software-Baustein (18) zugeordnet ist, der mit einem OLE-Automation- oder ATOMIC-Fernsteuerbaustein (19, 22, 27 bis 31, 35 bis 41) erweitert sind, damit die Vorrichtungen und Prozesse fernsteuerbar und die Limitierungen aufgehoben sind.



Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

BUNDESDRUCKEREI 10. 97 702 061/587

9/25

Beschreibung

Die Erfindung betrifft eine medizinische Systemarchitektur mit einer Modalität zur Erfassung von Bildern, einer Vorrichtung zur Verarbeitung der Bilder und einer Vorrichtung zur Übertragung der Bilder, bei dem die Vorrichtung zur Verarbeitung ein digitales Bildsystem mit einem Rechner aufweist, der nach einem Verfahren zum Datenaustausch zwischen verschiedenen Anwendungsprogrammen (OLE) mit grafischen Steuerelementen und einem Standard für OLE Custom Controls (OCX) arbeitet, wobei jedem einzelnen durch Grenzen limitierten Prozeß ein OCX-Software-Baustein zugeordnet ist.

Medizinische Systeme werden immer komplexer, während der Erweiterungsgrad medizinischer Systeme im gleichen Verhältnis anwächst. Dadurch wird jedoch eine sehr flexible Architektur benötigt.

Die bisher bekannten Architekturen sind im wesentlichen ohne dezentraler Software und Software-Bausteinen entworfen worden.

Weiterhin verhindert der einfache Einsatz von Class-Bibliotheken trotz des Gebrauchs von Objekt-Orientierung die Konstruktion von flexiblen und wiederverwendbaren Komponenten, da Classes in Wirklichkeit in großem Maße nicht richtig wiederverwendbar sind.

Die Erfindung geht von der Aufgabe aus, Software-Bausteine (Objekte) zu konstruieren, die ein Verhalten aufweisen, das sich möglichst selbst trägt. Weiterhin sollten die Verbindungen zwischen den Bausteinen im Verhältnis zum Ort dieser Bausteine (Objekte) unsichtbar sein, so daß sie entweder alle in einem Prozeß vereinigt oder über ein Netzwerk verteilt sein können.

Die Aufgabe wird erfindungsgemäß dadurch gelöst, daß die Systemarchitektur derart ausgebildet ist, daß die OCX-Software-Bausteine mit einem Fernsteuerbaustein erweitert sind, damit die Vorrichtungen und Prozesse fernsteuerbar und die Limitierungen aufgehoben sind. Durch die Kombination von Software-Bausteinen (Objekten) und flexibel verteilten Ereignismechanismen ergibt sich eine gute Lösung dieser Aufgabe.

Es hat sich als vorteilhaft erwiesen, wenn der Fernsteuerbaustein ein OLE-Automation-Interface ist und die Fernsteuerung nach dem OLE-Automation Standard erfolgt. Erfindungsgemäß kann der Fernsteuerbaustein ein Automation Interface Baustein ist.

Eine alternative vorteilhafte Lösung ergibt sich, wenn die Fernsteuerung mit Software-IC-Verbindungen beispielsweise nach dem ATOMIC-Standard erfolgt. Erfindungsgemäß kann in diesem Falle der Fernsteuerbaustein ein Connectable/Remote Interface Baustein ist.

Es hat sich als vorteilhaft erwiesen, wenn das Verfahren zum Datenaustausch der Microsoftstandard OLE und der Standard für OLE Custom Controls der Microsoftstandard OCX ist.

Erfindungsgemäß kann die medizinische Systemarchitektur Microsoft OCX zur Erstellung von Komponenten für grafische Benutzeroberflächen innerhalb eines Microsoft-Container-Prozeß verwenden, wobei Microsoft OCX mit OLE-Automation oder mit Software-ICs (Atomic) zur verteilten Ausbreitung eines Ereignisses mit der Kontrollebene und zwischen der Kontrollebene und der Serviceebene kombiniert werden kann.

Das wichtige neue Merkmal ist die Kombination von Microsoft OLE Custom Controls (OCX), einer neu entwickelten Software-Technologie, mit einem anderen generellen Microsoft-Scripting-Standard Interface, das als voll verteilungsfähiger netzwerkweiter Mechanis-

mus zur Ausbreitung eines Ereignisses (Event Propagation Mechanismus) dient, sobald Rechner-dezentralisiertes OLE erhältlich ist, um ein realistisches auf einer Systemarchitektur basierendes Model-View-Controller Konzept (MVC) zu erhalten.

Auch kann erfindungsgemäß Microsoft OCX zur Erstellung von Komponenten für grafische Benutzeroberflächen (Graphical User Interface (GUI) Components) innerhalb eines Microsoft-Container-Prozeß verwendet werden. Dadurch erhält man wirklich binärkompatible, wiederverwendbare GUI-Komponenten.

Weiterhin läßt sich erfindungsgemäß Microsoft OCX mit OLE-Automation oder mit Software-IC-Verbindungen zur verteilten Ausbreitung eines Ereignisses (Event propagation) mit lokaler Optimierung mit der Kontrollebene und auch zwischen der Kontrollebene und der Serviceebene kombinieren. Dies gibt die Flexibilität zur möglichen Verbreitung von Komponenten mit binärkompatiblen Interface, basierend auf Shared Libraries. Dadurch erhält man nicht GUI-abhängige und verteilbare Objekte.

Diese Näherung ermöglicht es, Prozesse nicht durch Design, sondern hauptsächlich durch Konfiguration von gemeinsam genutzten Programmkomponenten (Shared Libraries (DLLs)) zu erstellen.

Die Erfindung ist nachfolgend anhand von in der Zeichnung dargestellten Ausführungsbeispielen näher erläutert. Es zeigen:

Fig. 1 ein vernetztes Datenbanksystem gemäß dem Stand der Technik,

Fig. 2 bis 5 eine Darstellung zur Erläuterung der erfindungsgemäßen Zusammenarbeit der Software-Bausteine nach einem ersten Ausführungsbeispiel und

Fig. 6 eine Darstellung zur Erläuterung der erfindungsgemäßen Zusammenarbeit der Software-Bausteine nach einem zweiten Ausführungsbeispiel.

In Fig. 1 ist die Systemarchitektur eines medizinischen Computernetzwerkes dargestellt. Zur Erfassung medizinischer Bilder dienen die Modalitäten 1 bis 4, die als bilderzeugende Systeme beispielsweise eine CT-Einheit 1 für Computertomographie, eine MR-Einheit 2 für Magnetische Resonanz, eine DSA-Einheit 3 für digitale Subtraktionsangiographie und eine Röntgeneinheit 4 für die digitale Radiographie 4 aufweisen kann. An diese Modalitäten 1 bis 4 können Workstations 5 bis 8 angeschlossen sein, mit denen die Modalitäten 1 bis 4 gesteuert und die erfaßten medizinischen Bilder verarbeitet und abgespeichert werden können. Eine derartige Workstation ist beispielsweise ein sehr schneller Kleincomputer auf der Basis eines oder mehrerer schneller Prozessoren.

Die Workstations 5 bis 8 sind mit einem Bildkommunikationsnetz 9 zur Verteilung der erzeugten Bilder und Kommunikation verbunden. So können beispielsweise die in den Modalitäten 1 bis 4 erzeugten Bilder in einem zentralen Bildspeicher 10 abgespeichert oder an andere Workstations 5 bis 8 weit ergeleitet werden.

An dem Bildkommunikationsnetz 9 können weitere Workstations als Befundungskonsolen 11 und 12 angeschlossen sein, die mit einem lokalen Bildspeicher 13 und 14, beispielsweise einer Jukebox, verbunden sein können. In den Befundungskonsolen 11 und 12 können die erfaßten und im Bildspeicher 10 abgelegten Bilder nachträglich zur Befundung abgerufen und in dem lokalen Bildspeicher 13 und 14 abgelegt werden, von dem sie unmittelbar der an der Befundungskonsole 11 oder 12 arbeitenden Befundungsperson zur Verfügung stehen können.

Der Bild- und Datenaustausch über das Bildkommunikationsnetz 9 kann dabei nach dem DICOM-Standard erfolgen, einem Industriestandard zur Übertragung von Bildern und weiteren medizinischen Informationen zwischen Computern zur Ermöglichung der digitalen Kommunikation zwischen Diagnose- und Therapiegeräten unterschiedlicher Hersteller. An dem Bildkommunikationsnetz 9 kann ein Netzwerk-Interface 15 angeschlossen sein, über das das interne Bildkommunikationsnetz 9 mit einem globalen Datennetz verbunden ist, so daß die standardisierten Daten mit unterschiedlichen Netzwerken weltweit ausgetauscht werden können.

In der Fig. 2 ist ein erstes Beispiel einer Software-Architektur nach dem MVC-Konzept (Model-View-Controller) dargestellt, bei dem die Bausteine View, Control und Model jeweils in einem Prozeß enthalten sind. In einer dynamischen Link-Bibliothek (DLL) eines In-Process Servers 16, befindet sich der View-Bereich.

Eine dynamischen Link-Bibliothek (dynamic link library) ist eine Sammlung von Objektfiles, die zur Laufzeit (dynamisch) eines Prozesses diesem hinzugelinked (verfügbar gemacht) und von mehreren Prozessen gleichzeitig genutzt werden kann. Beim In-Process Server 16 ist die Implementierung eines Services derart realisiert, daß dieser Service immer im selben Prozeß ablaufen muß wie der Kunde (client), der diesen Service benötigt. Hierzu eignen sich insbesondere DLLs, da sie den Service dem Prozeß dynamisch bei Bedarf zur Verfügung stellen.

In dem Server 16 ist das Anwendungsprogramm 17 geladen, das mehrere OCX-Software-Bausteine 18 aufweist. An den OCX-Bausteinen 18 sind als Fernsteuerbausteine Automation Interface Bausteine 19 des OLE-Automation Standards angekoppelt. Aufgrund dieser Bausteine wird es ermöglicht, mit anderen lokalen Servern zu kommunizieren. So kann in einem zweiten lokalen Server 20 ein Controller 21 zur Kopplung der View-Ebene mit der Model-Ebene gespeichert sein. Dieser Controller 21 ist ebenfalls mit einem entsprechenden Automation Interface Baustein 22 gekoppelt.

Der Controller 21 eine Komponente aus der Model-View-Controller (MVC) Welt. Dabei repräsentiert der View eine Sichtweise auf das Model. Model und View sind über eine Controller Komponente gekoppelt. Diese Controller Komponente besteht zu wesentlichen Teilen aus einem Zustandsautomaten, einer sogenannten Finite State Machine (FSM).

Die Model Komponente kann in zwei weiteren lokalen Servern 23 und 24 enthalten sein, in denen die Service-Bausteine 25 und 26 über weitere Automation Interface Bausteine 27 bis 31 miteinander gekoppelt sind.

In den Fig. 3 bis 5 ist der Aufbau im wesentlichen der gleiche, lediglich die Prozeßaufteilung ist anders. Während im ersten Fall gemäß Fig. 2 alle MVC-Bausteine in jeweils einem getrennten Prozeß angeordnet sind, sind beim Beispiel gemäß Fig. 3 die Bausteine View und Control in einem Prozeß und die Model-Bausteine in getrennten Prozessen angeordnet. Das bedeutet, daß sich die Kontrollebene des Controllers 21 bereits in der gleichen dynamischen Link-Bibliothek (DLL) eines In-Process-Servers 32 wie das Anwendungsprogramm 17 der View-Komponente befindet.

Beim Gegenstand gemäß Fig. 4 sind die Bausteine Control und Model in einem Prozeß zusammengefaßt und befinden sich in der gleichen dynamischen Link-Bibliothek (DLL) des lokalen Servers 33. Erfindungsgemäß besteht auch die Möglichkeit, die Bausteine View, Control und Model gemäß Fig. 5 in einem Prozeß in der

dynamischen Link-Bibliothek (DLL) eines In-Process-Servers 34 zu vereinen, wobei ein weiterer Service-Baustein in einem weiteren lokalen Server 24 enthalten ist.

In der Fig. 6 ist ein zweites Beispiel nach dem MVC-Konzept (Model View Controller) dargestellt, das dem Beispiel gemäß Fig. 2 ähnlich ist. Auch hier befindet sich der View-Bereich wiederum in einer dynamischen Link-Bibliothek (DLL) des In-Process-Servers 16. An den OCX-Software-Bausteinen 18 des in dem Server 16 geladenen Anwendungsprogramm 17 sind sogenannte Software-IC-Verbindungen 35 (Connectable/Remote) als Fernsteuerbausteine anstelle der Automation Interface Bausteine 19 angekoppelt. Mittels dieser Bausteine wird die Kommunikation mit anderen lokalen Servern 20, 23 und 24 ermöglicht. So kann in dem lokalen Server 20 der Controller 21 zur Kopplung der View-Komponente mit der Model-Komponente sein. Dieser Controller ist ebenfalls mit entsprechenden Software-IC-Verbindungen 36 und 37 gekoppelt. Die Model-Komponente ist in den lokalen Servern 23 und 24 angeordnet, in denen über weitere Software-IC-Verbindungen 38 bis 41 die Service-Bausteine 25 und 26 enthalten sind.

Diese Connectable/Remote-Software-IC Fernsteuerbausteine sind voll-verteilungsfähige Eingangs/Ausgangs-Ereignisse, basierend auf ein Ereignis kommunizierendes Netzwerk, daß dynamisch linkbar und durch Eingangs/Ausgangs-Verbindungspunkte konfigurierbar ist.

Der Vorteil dieses erfindungsgemäßen Vorschlages liegt in seiner Flexibilität und noch mehr in seiner Produktivität zur Erlangung von in verschiedenen medizinischen Systemproduktarchitekturen wiederverwendbaren Software-Bausteinen.

Die Software-IC-Verbindungen erlauben eine wirkliche beliebige Verteilung der Komponenten in ablauffähigen Prozessen ohne in Verklemmungszustände zu geraten, wie dies bei anderen Kommunikationsmechanismen (z. B. Corba) der Fall sein kann, ohne Quellcode-Änderung. Die Komponenten können dadurch sogar zur Laufzeit beliebig kombiniert werden. Ein weiterer Vorteil ist, daß die Verbindungen n:m Verbindungen sein können, was bei herkömmlichen Systemen meist nicht möglich ist, wobei sich die Verbindungspartner anonym zur Laufzeit finden.

Zusätzlich unterscheiden sich die Komponentenverbindungen von herkömmlichen auch dadurch, daß auf den Verbindungen Ereignis-Daten übertragen werden und keine entfernten Methoden gerufen werden.

Patentansprüche

1. Medizinische Systemarchitektur mit einer Modalität (1 bis 4) zur Erfassung von Bildern, einer Vorrichtung (5 bis 8, 11, 12) zur Verarbeitung der Bilder und einer Vorrichtung (9) zur Übertragung der Bilder, bei dem die Vorrichtung (5 bis 8, 11, 12) zur Verarbeitung ein digitales Bildsystem mit einem Rechner aufweist, der nach einem Verfahren zum Datenaustausch zwischen verschiedenen Anwendungsprogrammen (OLE) mit grafischen Steuerelementen und einem Standard für OLE Custom Controls (OCX) arbeitet, wobei jedem einzelnen durch Grenzen limitierten Prozeß ein OCX-Software-Baustein (18) zugeordnet ist, dadurch gekennzeichnet, daß die Systemarchitektur derart ausgebildet ist, daß die OCX-Software-Bausteine (18) mit einem Fernsteuerbaustein (19, 22, 27 bis 31, 35 bis 41) erweitert sind, damit die Vorrichtungen und

Prozesse fernsteuerbar und die Limitierungen aufgehoben sind.

2. Medizinische Systemarchitektur nach Anspruch 1, dadurch gekennzeichnet, daß der Fernsteuerbaustein ein OLE-Automation-Interface (19, 22, 27 bis 31) ist. 5

3. Medizinische Systemarchitektur nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß die Fernsteuerung nach dem OLE-Automation Standard erfolgt. 10

4. Medizinische Systemarchitektur nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, daß der Fernsteuerbaustein ein Automation Interface Baustein (19, 22, 27 bis 31) ist.

5. Medizinische Systemarchitektur nach Anspruch 1, dadurch gekennzeichnet, daß die Fernsteuerung mit Software-IC-Verbindungen (35 bis 41) erfolgt. 15

6. Medizinische Systemarchitektur nach Anspruch 1 oder 5, dadurch gekennzeichnet, daß die Fernsteuerung nach dem ATOMIC-Standard erfolgt. 20

7. Medizinische Systemarchitektur nach Anspruch 5 oder 6, dadurch gekennzeichnet, daß der Fernsteuerbaustein ein Connectable/Remote Interface Baustein (35 bis 41) ist.

8. Medizinische Systemarchitektur nach einem der Ansprüche 1 bis 7, dadurch gekennzeichnet, daß das Verfahren zum Datenaustausch der Microsoftstandard OLE ist. 25

9. Medizinische Systemarchitektur nach einem der Ansprüche 1 bis 8, dadurch gekennzeichnet, daß der Standard für OLE Custom Controls der Microsoftstandard OCX ist. 30

10. Medizinische Systemarchitektur nach einem der Ansprüche 1 bis 9, gekennzeichnet durch die Verwendung von Microsoft OCX zur Erstellung von Komponenten für grafische Benutzeroberflächen innerhalb eines Prozeß-Microsoft Containers. 35

11. Medizinische Systemarchitektur nach einem der Ansprüche 1 bis 10, gekennzeichnet durch die Kombination von Microsoft OCX mit OLE-Automation zur verteilten Ausbreitung eines Ereignisses mit der Kontrollebene und zwischen der Kontrollebene und der Serviceebene. 40

12. Medizinische Systemarchitektur nach einem der Ansprüche 1 bis 11, gekennzeichnet durch die Kombination von Microsoft OCX mit Software-IC-Verbindungen zur verteilten Ausbreitung eines Ereignisses mit der Kontrollebene und zwischen der Kontrollebene und der Serviceebene. 45

Hierzu 6 Seite(n) Zeichnungen 50

55

60

65

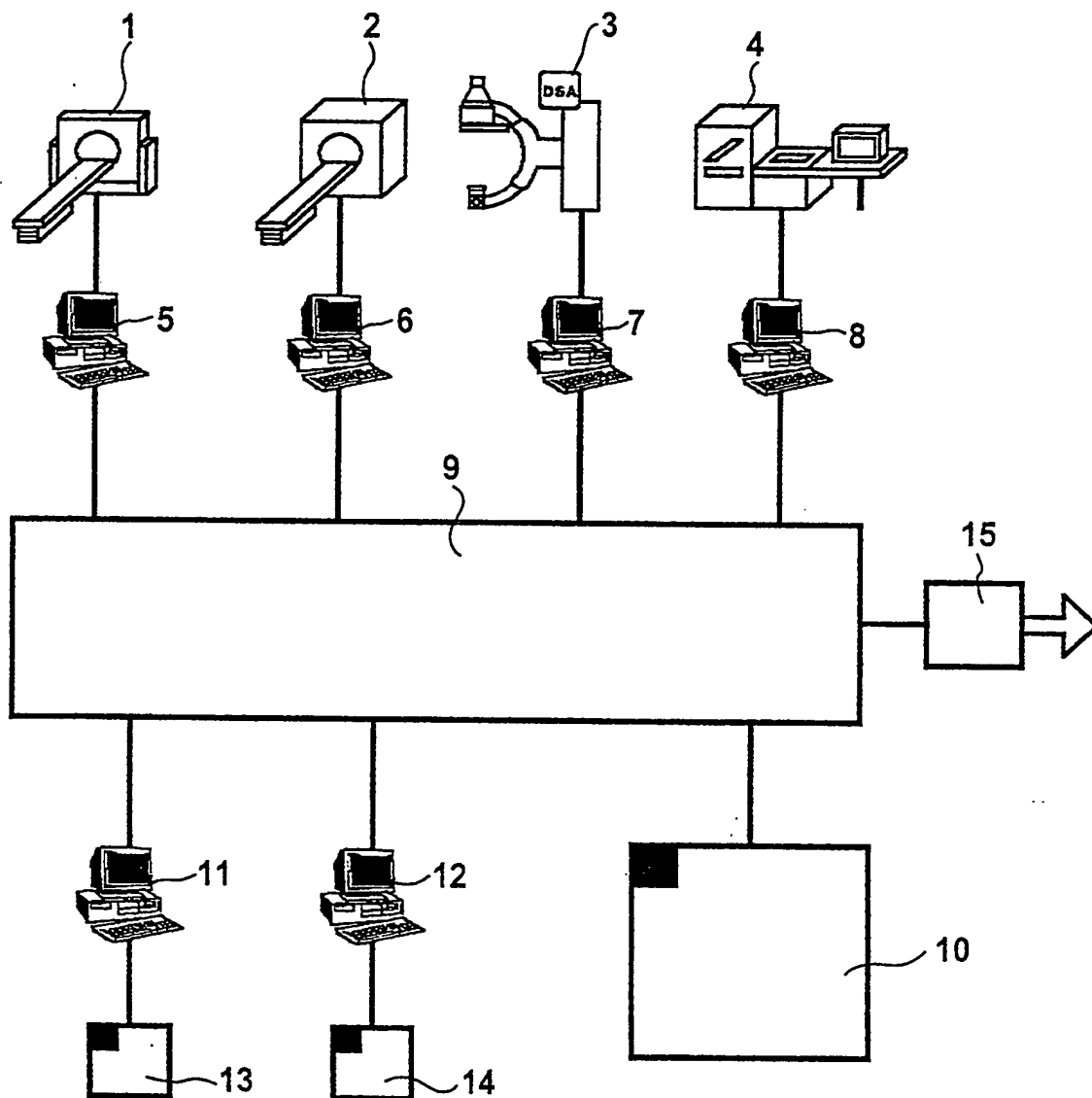


FIG 1

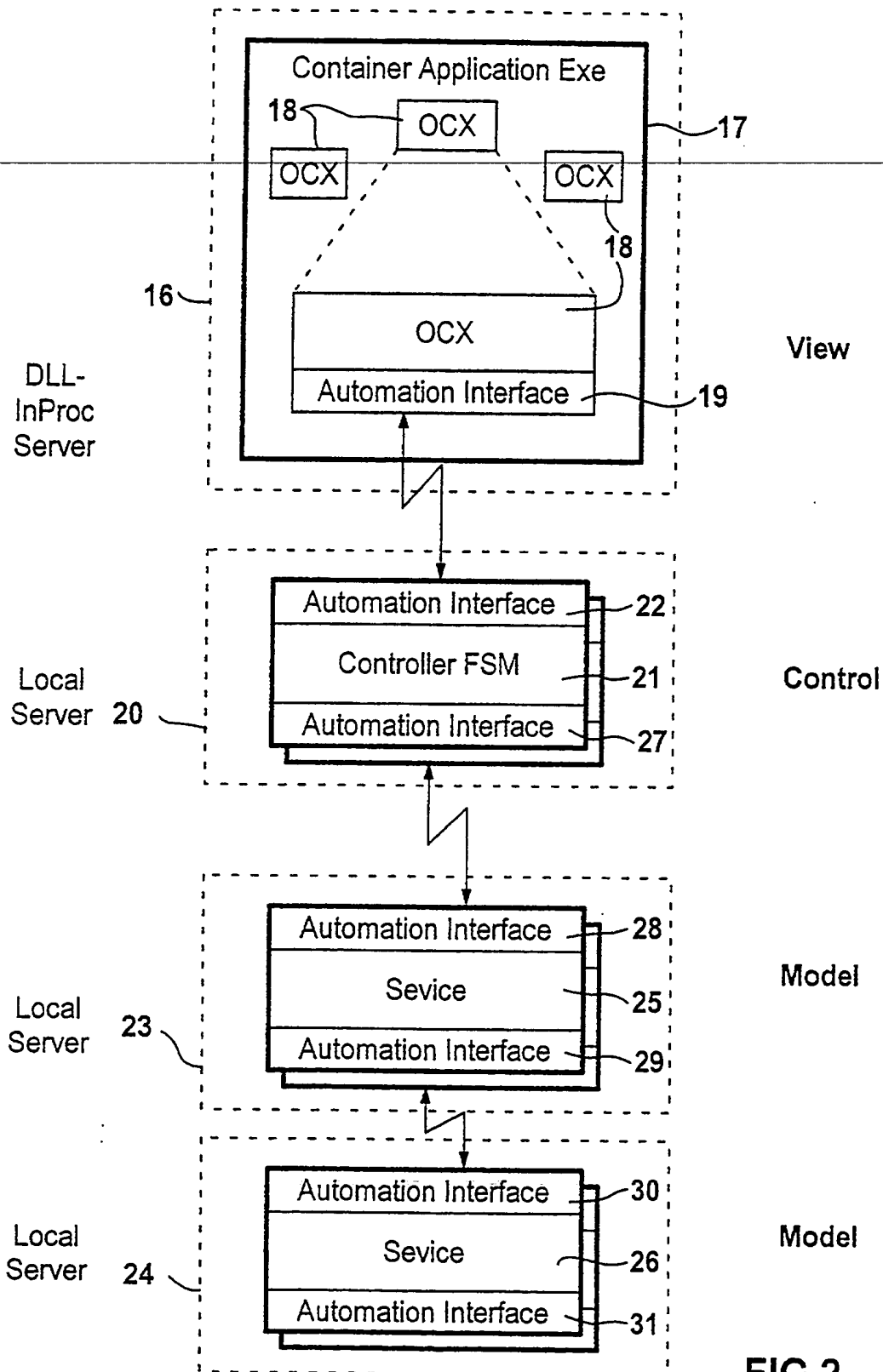


FIG 2

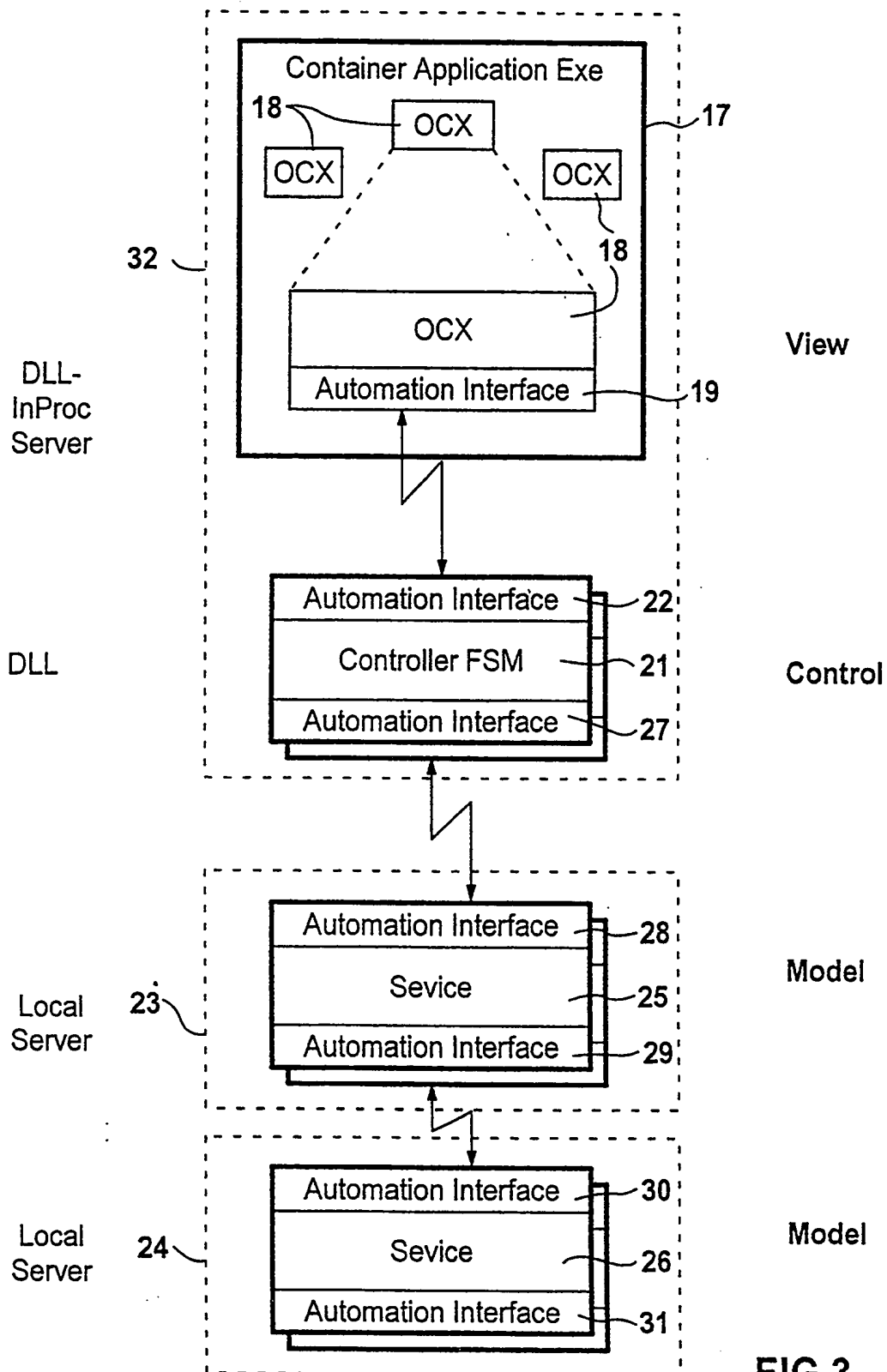


FIG 3

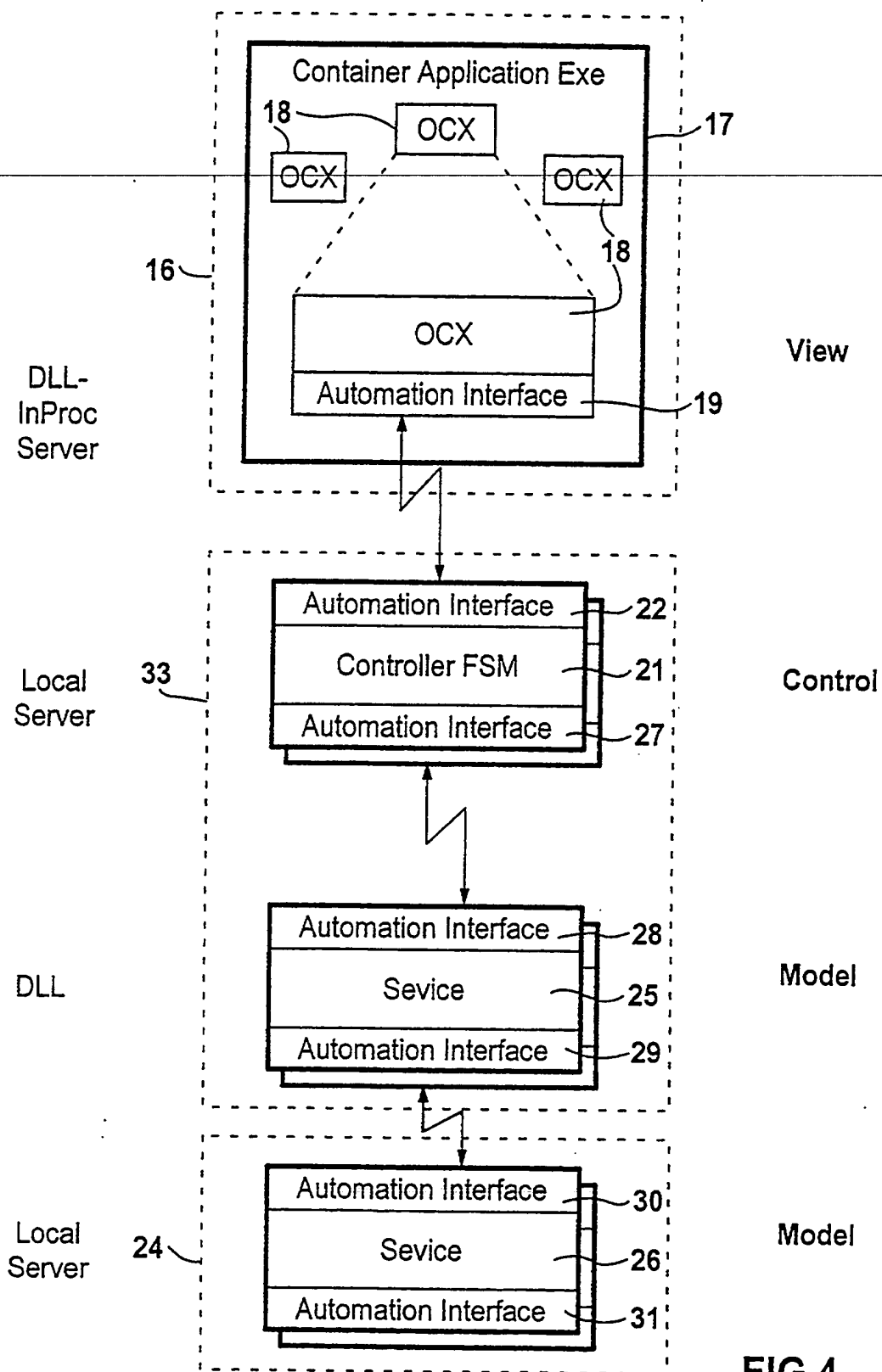


FIG 4

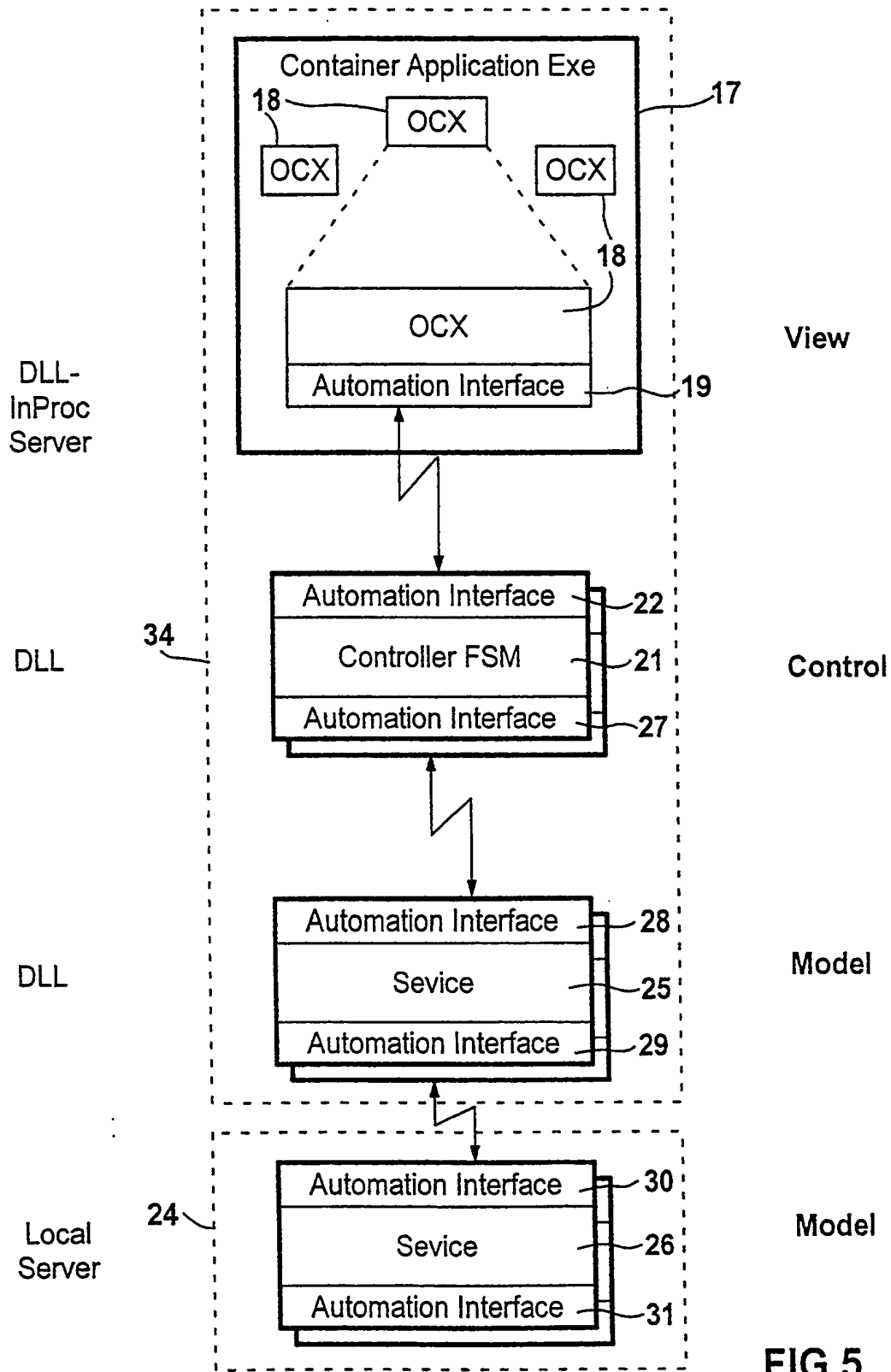


FIG 5

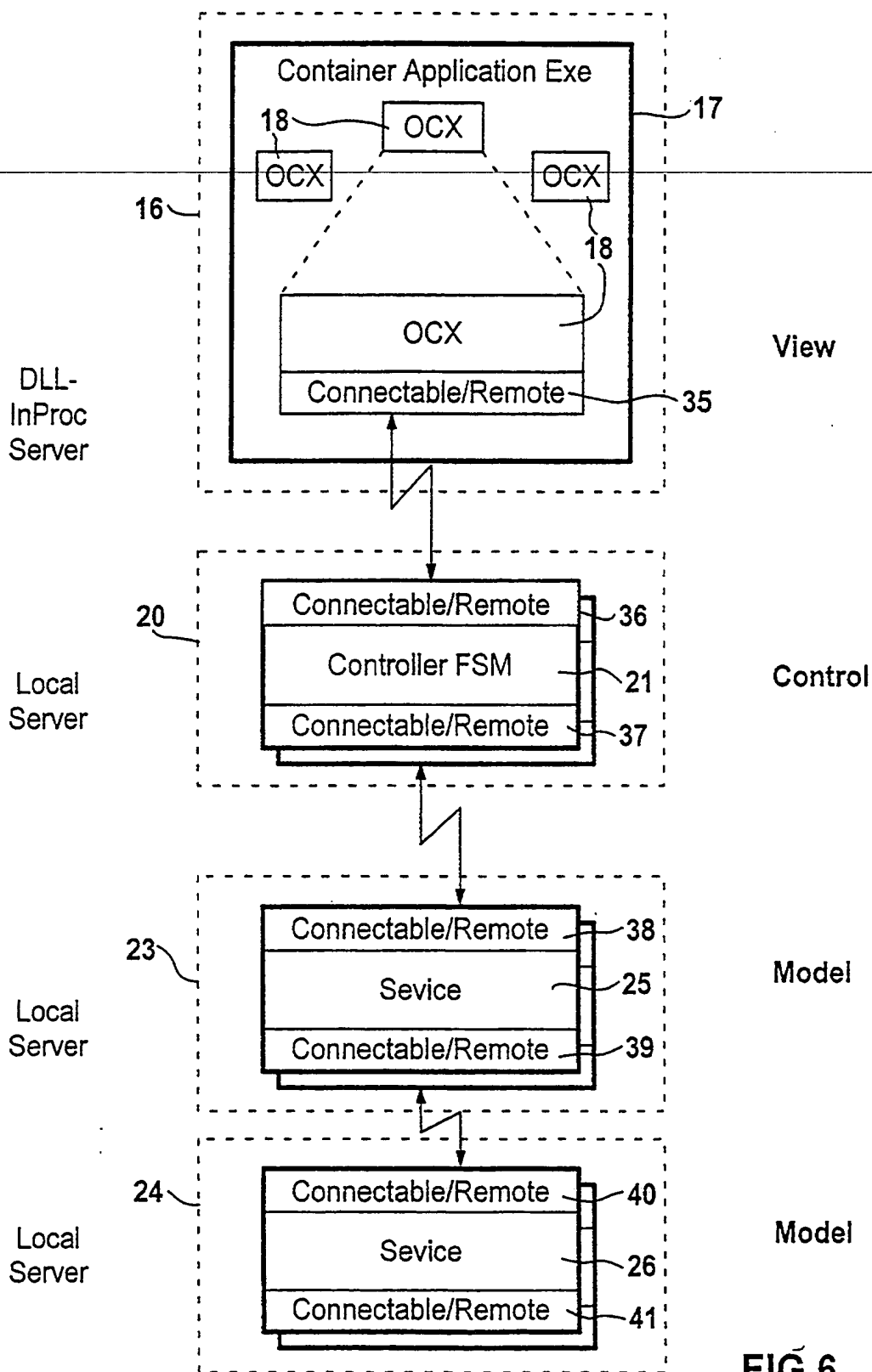


FIG 6